# GesturePIN: Using Discrete Gestures for Associating Mobile Devices

Ming Ki Chong
Computing Department
Lancaster University
Lancaster, UK
chong@comp.lancs.ac.uk

Gary Marsden
Department of Computer Science
University of Cape Town
Cape Town, South Africa
gaz@cs.uct.ac.za

Hans Gellersen
Computing Department
Lancaster University
Lancaster, UK
hwg@comp.lancs.ac.uk

## ABSTRACT
Mobile devices with wireless network capabilities can be associated to form ad hoc networks to share resources; however, such an association of devices requires authentication. At present, PIN is the common authentication method, but in many cases, small devices may not have input interfaces to accommodate PIN entry. We therefore design a gesture-based authentication scheme, called *GesturePIN*, for associating multiple mobile devices; our solution provides the advantage of being adaptable to any PIN authentication systems. We have also conducted a quantitative user study to understand the speed and accuracy of people using our gesture-based system compared to using PIN.

## Categories and Subject Descriptors
D.4.6 [**Operating Systems**]: Security and Protection – Authentication; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Input devices and strategies*.

## General Terms
Design, Security, Human Factors.

## Keywords
Spontaneous interaction, device association, device authentication, gesture password.

## 1. INTRODUCTION
Mobile devices, such as cellular phones, portable music players, etc., are becoming pervasive. With the rapid adoption of mobile devices, users are expected to encounter a frequent task of setting up communication channels dynamically to exchange data; in other words, device associations[1] are expected to happen often and spontaneously. Wireless ad hoc networks provide suitable environments for such events. However, due to the inherent insecurity of wireless communication, wireless channels give rise to security problems [5, 8], like man-in-the-middle (MITM) attacks [3]. To overcome the security problems, users must be involved to assist the authentication of device associations [3].

Following the trend of growth of mobile technology, capabilities (like processing speed, storage capacity, etc.) of mobile devices are improving continuously. Although the specifications are advancing, one likely to remain the same is the small sizes of mobile devices. This requirement is necessary for mobility, as mobile devices must remain small enough that users are willing to carry them [6]. However, this brings a physical size limitation to the input and output (I/O) capabilities, such as small (or no) output displays and slow input methods and, as a result, mobile devices have limited user interfaces (UI). The limited UI capabilities of mobile devices induce a challenging task for user-assisted authentication, especially in ubiquitous computing. A common technique for authenticating devices is to enter the same PIN code into the involved devices, like a Bluetooth "passkey" for pairing, but such scheme requires the devices to have a number input interface. Particularly, smaller and specialized devices (like GPS loggers, portable MP3 players, etc.) do not have interfaces to accommodate number entries, and, hence, PIN authentication becomes impractical as it cannot be implemented. To overcome this problem, an alternative authentication technique is required.

In recent years, accelerometers have been increasingly integrated into mobile devices (e.g. mobile phones and portable media players). As accelerometers are small in size, they could be embedded within mobile devices without affecting their overall features. A built-in accelerometer allows a device to sense user's movements and, as a result, it provides a modality for user input. As more uses of accelerometers are being discovered, we confidently predict that mobile devices will be equipped with built-in accelerometers. In this paper, our work is based on the use of accelerometers to detect users' movements as inputs for authenticating mobile devices.

### 1.1 Organization
We first present the motivation and the design of our work. We explore a new gesture-based authentication method for user-assisted association of mobile devices, specifically for devices with limited UI, and we explain how our design can be built to work with PIN authentication protocols. We then present the results and a discussion of our initial study of the system before conclusions are drawn and future directions for research are identified.

[1] The term "*association*" refers to the set up of a connection among multiple devices. In many publications, the term "*pairing*" is often used. In this paper, pairing is referred to an association of a *pair* of devices, while association has the cardinality of *two or more* devices.


*MobileHCI'10*, September 7–10, 2010, Lisbon, Portugal.
ACM 978-1-60558-835-3/10/09.

## 2. MOTIVATION

There are two basic types of authentication: (i) *device authentication* and (ii) *user authentication*. The former refers to a device proving its existence to its connecting peer device(s) (like device pairing), and the latter refers to a user proving his/her claimed identity to a system that he/she is the genuine individual. In this work, our interest is in the use of gesture movements to achieve device authentication.

### 2.1 Background

Two publications ([6] and [5]) inspired the work described in this paper.

In [6], Patel *et al.* suggest a gesture-based authentication scheme for pairing a mobile device with an untrusted public terminal using a challenge-response protocol. Their use-case scenario describes a user who walks up to a public display and authenticates his/her handheld device to the display. First, the system generates a random series of shakes and pauses as a gesture sequence (the challenge), and then the user authenticates his/her device by mimicking the gestures of the sequence (the response). Their system is suitable for situations when a situated terminal with a display output is available at one end of the association; however, the scheme cannot be used when only mobile devices are involved. Often, device associations happen spontaneously; there may not be a display terminal accessible during the associations. Furthermore, when multiple devices are connecting to a terminal at once, the sharing of the terminal display could be problematic.

In [5], Mayrhofer and Gellersen suggest a method which uses accelerometer data to pair two mobile devices. Their method requires a user to hold the pairing devices tightly together and shake the devices for a short period. As the devices are shaken, acceleration data is captured and analysed; afterward, the devices will only be paired if the readings of the accelerometers are similar. The advantage of this method is that shaking is an easy and intuitive gesture as it requires little learning and cognitive demand [4, 5]. However, the interaction restricts the number of associating devices, as only one user can hold and shake the connecting devices at a time. Depending on the sizes of the devices, the user cannot hold too many of them at once. Consequently, the method is not feasible for situations of forming a large-scale network of numerous devices; for instance, a group of gamers cannot shake their gaming devices to form an ad hoc network. In addition, since only one user can operate the shaking, other users are required to surrender their physical possession of their devices during authentication. In some social contexts, this may not be acceptable or appropriate; e.g. associating devices with strangers.

Although [6] and [5] suggest solutions of using movements for pairing a mobile device to a terminal or to another mobile device, both solutions have problems when the cardinality of devices increases. So far, most of the research done on device association is based on pairing of two devices, and we found no existing solution that explores device authentication of a group of UI limited devices.

## 3. DESIGN

In this section, we present a design solution called *GesturePIN*. It uses movement gestures for associating a group of devices.

### 3.1 Input Devices

As described previously, the aim of this work is to design for mobile devices with built-in accelerometers; we therefore assume all mobile devices have the facility to capture gestures.

When associating devices, often the devices have different I/O capabilities. For example, some devices have small display output, while others may only have a speaker, a vibration motor, or LEDs. We therefore classify mobile devices according to their I/O capabilities (see table 1). Devices with text or touch input can use the standard PIN authentication technique to pair/associate devices. We are particularly interested in devices with less input capabilities, such as gestures only and simple buttons. We have no particular preference for the type of output; the only requirement is the output must be able to emit differentiable binary signals to the users (e.g. beeping/vibrating/light flashing) to indicate a success of a failure.

**Table 1. Classification of I/O capabilities of mobile devices**

| **Input** (*all devices are assumed to have built-in accelerometers*) | **Output** |
|---|---|
| Gestures only | Binary (*binary signals, e.g. beeps, LED flashes, vibrations, etc.*) |
| Simple buttons (*for basic controls only*) + Gestures | Text (*alphanumeric characters*) |
| Text (*e.g. a key pad*) + Gestures | Multimedia (*voice + tunes, images, videos*) |
| Touch (*e.g. a touch screen*) + Gestures | |

### 3.2 Authenticator

Similar to the work in [6], our system uses gesture sequences as passkeys. We adopt *discrete gesture password* [2] (shown in figure 1) as the gesture elements of our authenticator. The ten gesture elements[2] are 3D stroke-based directional movements. The elements are based on the spatial orientation of the input device, and they were designed with the intention that each gesture has a symmetrical gesture in the mirror direction (we refer readers to [2] for an in-depth discussion of gesture password).

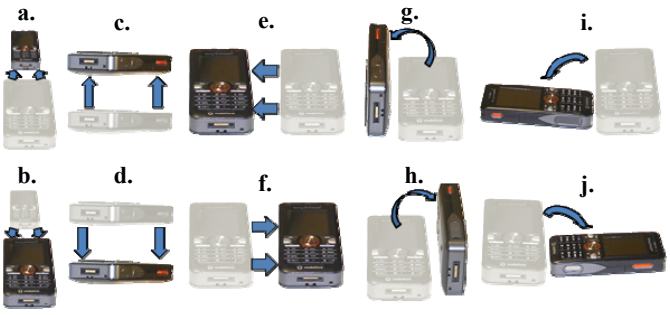


**Figure 1. Ten gesture password elements of GesturePIN. (a) Forward, (b) Backward, (c) Up, (d) Down, (e) Left, (f) Right, (g) Tilt Left, (h) Tilt Right, (i) Swing Left, (j) Swing Right**

[2] Go to http://goo.gl/vWfK for a video illustration of the ten gestures.

## 3.3 Device Association Scheme

For simplicity of demonstration, we adopt the following scenario to illustrate our design idea.

*John has a video about his recent vacation uploaded on his portable media player. He goes to a cafe to meet his friends and to show them the video. After viewing, everyone is very keen to have a copy of the video, so John decides to share it. Since there are many devices, it would be time consuming to share the file individually, so, instead, John initiates a wireless ad hoc network to share the file with everyone at once. However, some of the devices do not have a number input interface, so PIN authentication cannot be done. Instead, GesturePIN is used to authenticate the devices. To do so, the group performs the following 3-stage procedure:*

**Stage 1**: *Devices* A*wakening*. To start, John and his friends, individually, shake their devices rapidly and randomly; if the device has a simple button interface, the user pushes it as the wake button. This action awakens the devices and informs them to start a group connection and to find the network identifiers of one another. When each device is ready, it outputs a simple signal (e.g. a beep or a pulse of vibration) to indicate it is ready.

**Stage 2**: *Passkey Sharing*. John, the initiator, selects and demonstrates a random sequence of gestures (i.e. the passkey), and his friends follow his gesture sequence. Since John's device has an output display, his device translates the input gestures into a PIN code. John shares the code with those who have the devices with number input, so they can use PIN authentication instead.

**Stage 3**: *Confirming*. After entering the passkey, similar to stage 1, if the device has a simple button interface, the user presses the button (e.g. an "OK" button) to confirm the passkey entry. Else, the user shakes the device vigorously to indicate confirmation. Lastly, the devices output a signal to acknowledge the user input.

*Upon a successful authentication, where everyone has entered the same passkey correctly, an ad hoc network is formed, and now, John can share his video with his friends.*

## 3.4 Gestures to PIN Codes Translation

The ten gestures in figure 1 are compatible with PIN digits; they can be translated into each other, like *forward = "1"*, *back = "2"*, and so forth. As a result, the gesture scheme can be implemented on top of any standard PIN entry authentication protocols, e.g. Bluetooth pairing, MANA III protocol [3, 9]. A device with only gesture input can now authenticate with devices that only have number input; however, the restriction is that the gesture input device must have a number display (see figure 2 for an illustration).

In addition, the 3-stage device association scheme (in section 3.3) is not only limited to the suggested gestures in figure 1. The concept of GesturePIN can adopt any types of gestures, as long as the gestures are discrete and translatable to PIN digits. For example, the shapes of the PIN digits can be drawn as gestures; thus, users can write PINs in midair instead of pressing buttons.

## 3.5 Security

Gesture authentication is similar to PIN authentication. When a user authenticates his/her device, the user is required to produce

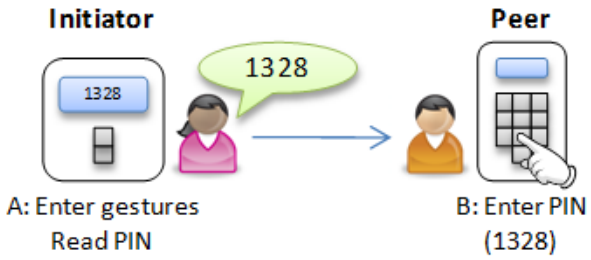


**Figure 2. Alice (A) enters gestures into her device. The device translates the gestures into a PIN code (1328) and displays the code. Alice reads the code to Bob (B), and Bob enters the code into his device using the number keypad.**

the correct passkey (a sequence of numbers/gestures) given by the initiator.

Small and specialised devices have limited input. Without number input, the current authentication method is limited to either using a default passkey or no passkey. Our solution enables users the flexibility to select their own passkey, and hence, the key is not fixed. By disclosing the chosen passkey only to the legitimate peers (via the human channel), an end-to-end authentication is assured; thus, preventing impersonation attacks.

One drawback of using movements is that the system is susceptible to shoulder surfing attacks. Due to the nature of using movements as inputs, an attacker can observe gesture entries [2]. Anyhow, our system is designed for transient interactions, like file sharing. Once a connection terminates, the passkey is discarded; hence, a different passkey is used for each new session. An assailant needs to be near the users during the authentication to attempt an attack. Besides, the gestures we adopted are based on arm movements; faster and more discreet gestures that exploit more flexible muscles could be used instead.

# 4. USER STUDY

A quantitative user study was conducted to compare the differences in performance between PIN and gesture password. The study records the speed and accuracy of password entries. Entry speed is defined as the time (measured in seconds) users took to enter a valid password, while accuracy is defined as the number of attempts used. 18 learners (aged 17 to 33, mean=24.4) from a skills training centre in Khayelitsha, Cape Town, were recruited as participants. The study was conducted as a repeated measures study with each participant exposed to both testing systems (PIN and gesture password).

A PIN authenticator was implemented to log entry time and number of tries. The authenticator was implemented as a Java MIDlet running on a Sony Ericsson V630i mobile phone. The study was conducted without the use of a real gesture password prototype. For simplicity and as a proof of concept, the *Wizard of Oz* prototyping technique [7] was adopted (discrete gesture password can be implemented as a real prototype, we refer readers to [1] for details). The results of each gesture entries were recorded by the experiment conductor. A rectangular object (we used a mobile phone) was given to the subject to represent a sensor device.

At first, all participants undertook a training session (trained by the experiment facilitator) to familiarize themselves with the test-

ing systems. All of the subjects were familiar with the concept of a PIN beforehand; however, since gesture password is a new concept, the movements of the ten gestures required explanation and illustration, and the subjects were trained until they decided they were comfortable using the systems. After training, each subject was given a piece of paper printed with five PINs and five gesture passwords that were randomly chosen, and the subjects' task was to enter the given passwords. At the end of the study, 90 PINs and 90 gesture passwords were entered. However, two entries of gesture passwords were discarded due to difficulties (after three attempts with each of those passwords, the users still failed to enter the gestures correctly).

The results of the time of entries show that on average, the speed of entering a valid PIN ($M$=5.74, $SE$=0.23) was significantly faster than entering a gesture password ($M$=9.53, $SE$=0.41), $t(87)=8.954$, $p<0.001$, $r=0.69$.

The measure of accuracy was classified as the subjects using either (i) *first attempt* or (ii) *multiple attempts* to enter their given passwords correctly. We recorded 86 first-attempts and 4 multiple-attempts for PINs and 48 first-attempts and 40 multiple-attempts for gesture passwords. A chi-square test showed there is a significant difference ($\chi^2=40.21$, $df=1$, $p<0.001$) in accuracy between PINs and gesture passwords.

## 4.1 Discussion

The results from the study show that users' performance in using PINs for authentication is significantly superior to using gesture passwords in both speed and accuracy. The results are not surprising since the participants were previously familiar with PIN but not with gesture password. The subjects' knowledge of PINs can influence their performance, as they already have a strategy to cope with PIN. We acknowledge the results of our study are biased because the subjects had prior experience with PIN; however, we believe it is more realistic to use subjects from this population as mobile phones have already been adopted pervasively.

Although the finding shows PIN is significantly superior, the means of the time of entries is different by merely a few seconds (|9.53–5.74|=3.79s). The mean time of gesture password entries shows people enter gestures in a relatively short amount of time. We consider the average of 9.53s is acceptable for transient associations (e.g. the scenario in section 3.3). However, many failed first-attempts were recorded from the gesture entries. For this reason, we realized a scheme for correcting wrong gesture entries is needed. For example, if an incorrect gesture is entered, the user can draw a shape (like a circle or a gesture other than the predefined gestures) to remove the entry. The concept is similar to the backspace key (on a standard computer keyboard) for text entry.

Bearing in mind that the study was the subjects' first experience with gesture password, if a longer period (like days or weeks) was given to practise the gestures, the performance of gesture password entries may improve and be comparable to PIN.

Furthermore, since device authentication is not knowledge-based, it relieves the users from the burden of recalling passwords [6]. Other than device authentication, it is possible to adopt gesture password for user authentication; however, it is not recommendable, as the results from [2] show gesture passwords are not easily memorable.

# 5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented GesturePIN, a solution that adopts discrete gesture password for associating multiple devices, especially for devices with limited input capability. Our design has the advantages of (i) not restricting the number of associating devices (the cardinality can go up to *any* number of devices) and (ii) being able to adapt to work with any PIN authentication systems (since discrete gestures can be translated into numbers). An initial user study was conducted to compare the speed and accuracy between PIN and gesture password entries. The results of the study show users performed better in entering PIN; whilst PIN may be superior in some ways, the study shows gesture password is at least usable, thus it is viable for devices with limited UI to adopt GesturePIN. However, further study is needed to discover the full potential of gesture password. For future work, a fully functional prototype of GesturePIN may yield valuable findings, as the prototype can be used for investigating qualitative measures like users acceptance, their opinions and the quality and the potential uses of the system perceived by the users.

# 6. REFERENCES


[1] Chong, M. K. 2009. Usable authentication for mobile banking. MSc Thesis, University of Cape Town.

[2] Chong, M. K., and Marsden, G. 2009. Exploring the use of discrete gestures for authentication. In Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction (Uppsala, Sweden). INTERACT '09. Springer-Verlag, Heidelberg, 205–213.

[3] Gehrmann, C., Mitchell, C. J., and Nyberg, K. 2004. Manual authentication for wireless devices. RSA CryptoBytes, 7 (1), 29-37.

[4] Holmquist, L. E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., and Gellersen, H.-W. 2001. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In Proceedings of the 3rd international conference on Ubiquitous Computing. UbiComp '01. Springer-Verlag, 116-122.

[5] Mayrhofer, R., and Gellersen, H. 2009. Shake well before use: Intuitive and secure pairing of mobile devices. IEEE Trans. Mob. Comput. 8 (6), 792-806.

[6] Patel, S. N., Pierce, J. S., and Abowd, G. D. 2004. A gesture-based authentication scheme for untrusted public terminals. In Proceedings of the 17th annual ACM symposium on User interface software and technology (Santa Fe, NM, USA). UIST '04. ACM Press, New York, NY, 157-160. DOI= http://doi.acm.org/10.1145/1029632.1029658

[7] Sharp, H., Rogers, Y., and Preece, J. Interaction Design: Beyond Human-computer Interaction, 2nd edition, John Wiley & Sons, 2007.

[8] Stajano, F., and Anderson, R. 1999. The resurrecting duckling: Security issues for ad-hoc wireless networks. In Security Protocols: 7th International Workshop. Springer-Verlag, 172-182.

[9] Valkonen, J., Asokan, N., and Nyberg, K. 2006. Ad hoc security associations for groups. In Security and Privacy in Ad-Hoc and Sensor Networks. ESAS 2006. Springer-Verlag, 150-16